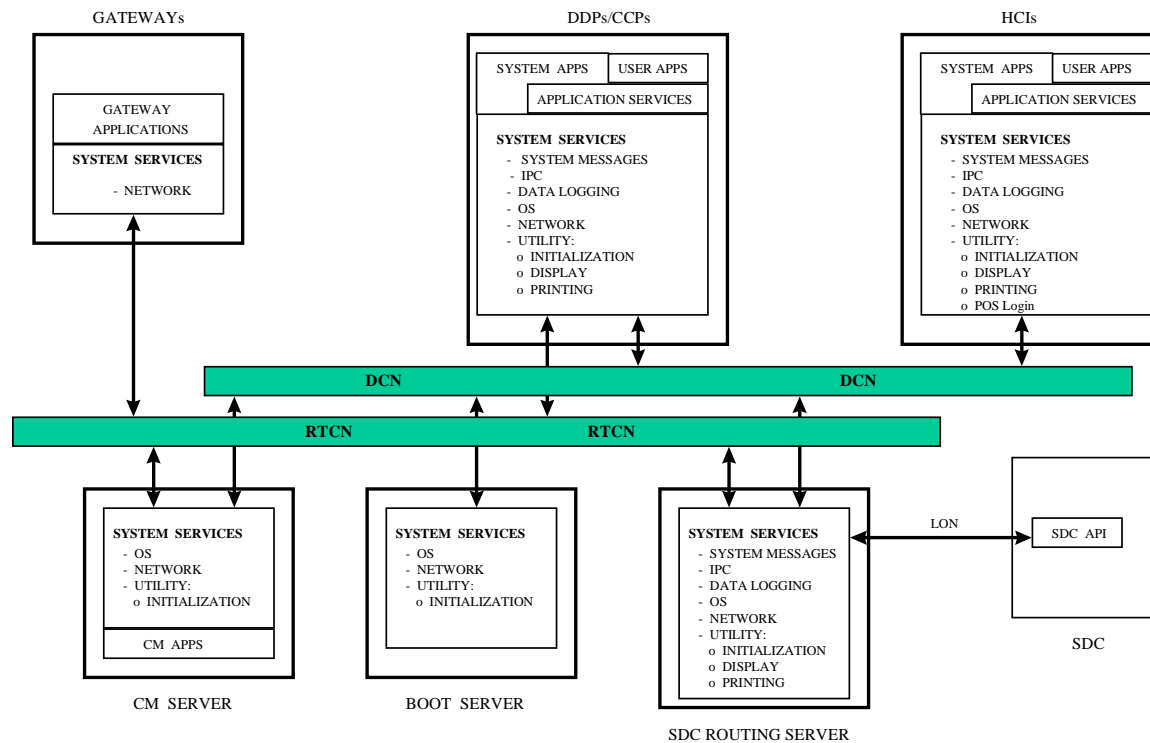


## 1. Operating System

### 1.1 Operating System Introduction

#### 1.1.1 Operating System Overview

The Operating System CSC provides a stable development and operational platform to execute custom/commercial code in realtime. The Operating System CSC consists of generating and configuring the Operating System baseline image as well as the distribution of that image to the client hardware.



SYSTEM SERVICES OVERVIEW

#### 1.1.2 Operating System Description

Operating Systems provides a method of building , managing, distributing, updating, and tuning operating systems for the various platforms and computing bases. The following categories comprise the basic requirements of the Operating Systems functionality:

- General
- Operating System Performance Management
- Operating System Server Configuration
- Operating System Anomaly Management

## 1.2 Operating System Specifications

### 1.2.1 Operating System Groundrules

The following is a list of groundrules and assumptions that relate to the Operating System CSC for Redstone:

- **TCID and SCID will be resident on local disk.**
- **Network Information Services (NIS) will not be required in Operational set.**
- **Processor locking and realtime priorities will be utilized.**
- **Kernel tuning will be implemented.**
- 100MB Ethernet will be used to load the operating system from the Boot server to all RTPS platforms.
- CM baselines will be in tape (4MM DAT) format.
- NRS will provide host table manipulation. The Operating System will only provide the minimum network configuration to install the client machines.

## 1.2.2 Operating System Functional Requirements

The Functional Requirements for CSC Name are arranged in the following major/minor functions:

1. General
2. Performance Management
3. Server Configuration
4. Anomaly Management

### 1. General

- 1.1 A capability will be provided to maintain current, tested, and functionally proven baselines, for CLCS associated platforms.
- 1.2 Operating system builds will be developed and supplied to CM.
- 1.3 The operating systems will allow systems administrators to have autonomy and authority to perform any system upgrades, additions, deletions, and modifications.
- 1.4 A capability will be provided to maintain system documentation and user information including, but not limited to:
  - a) USER PASSWORD FILES
  - b) SYSTEM GROUP FILES
  - c) OPERATING SYSTEM CONFIGURATION MANAGEMENT FILES
  - d) SERVER CONFIGURATIONS
  - e) CLIENT CONFIGURATIONS
  - f) OPERATING SYSTEMS DOCUMENTATION
- 1.5 A method and procedures will be provided for full and incremental backups of Operating System to facilitate system recovery.
- 1.6 **Implement back-up procedures for Net and Boot server in a set.**
- 1.7 **Initial login screen will contain the SCID and TCID identifiers for Development Platform load.**
- 1.8 **Create and Operational and Development load.**
- 1.9 Perform tuning and optimization necessary for the reliable and continuous functionality of meeting developer/user requirements.

### 2. Performance Management

- 2.1 Operating Systems performance will be tunable based on hardware and system workload for each environment.
- 2.2 Operating Systems tuned parameters should be portable and should maintain the ability to function irrespective of network media.
- ~~2.3 Operating systems performance will direct server and client optimization parameters for each environment and throughout each phase of project.~~
- ~~2.4 Operating Systems performance will evolve as network configurations evolve throughout the development phases of the project.~~

### 3. Server Configuration

- 3.1 Server OS Configuration will facilitate the following (either using dedicated equipment or in combined services):

# Software Requirements and Design Specification Template

- a) BOOT FUNCTION – Remote Installation Services
  - b) NETWORK FUNCTION – Network Information Services
  - c) READ/WRITE FUNCTION – Data Repository
  - d) NTP FUNCTION -- Network timing services
  - e) NFS FUNCTION – Network file system services
  - f) ~~EMAIL FUNCTION – Electronic mail services~~
  - g) ~~ACTIVITY MANAGER – Systems Activities management services~~
- 3.2 Operating Systems Server Configuration will be tuned to the most optimized configuration possible to make use of the available hardware and software resources.
- ~~3.3 Operating Systems server configuration will be molded into the overall scheme of the operating system configuration management design.~~

## 4. Anomaly Management

- 4.1 An Operating Systems Anomaly/Assistance Utility will be made available to the users for problem reporting, resolution and support.

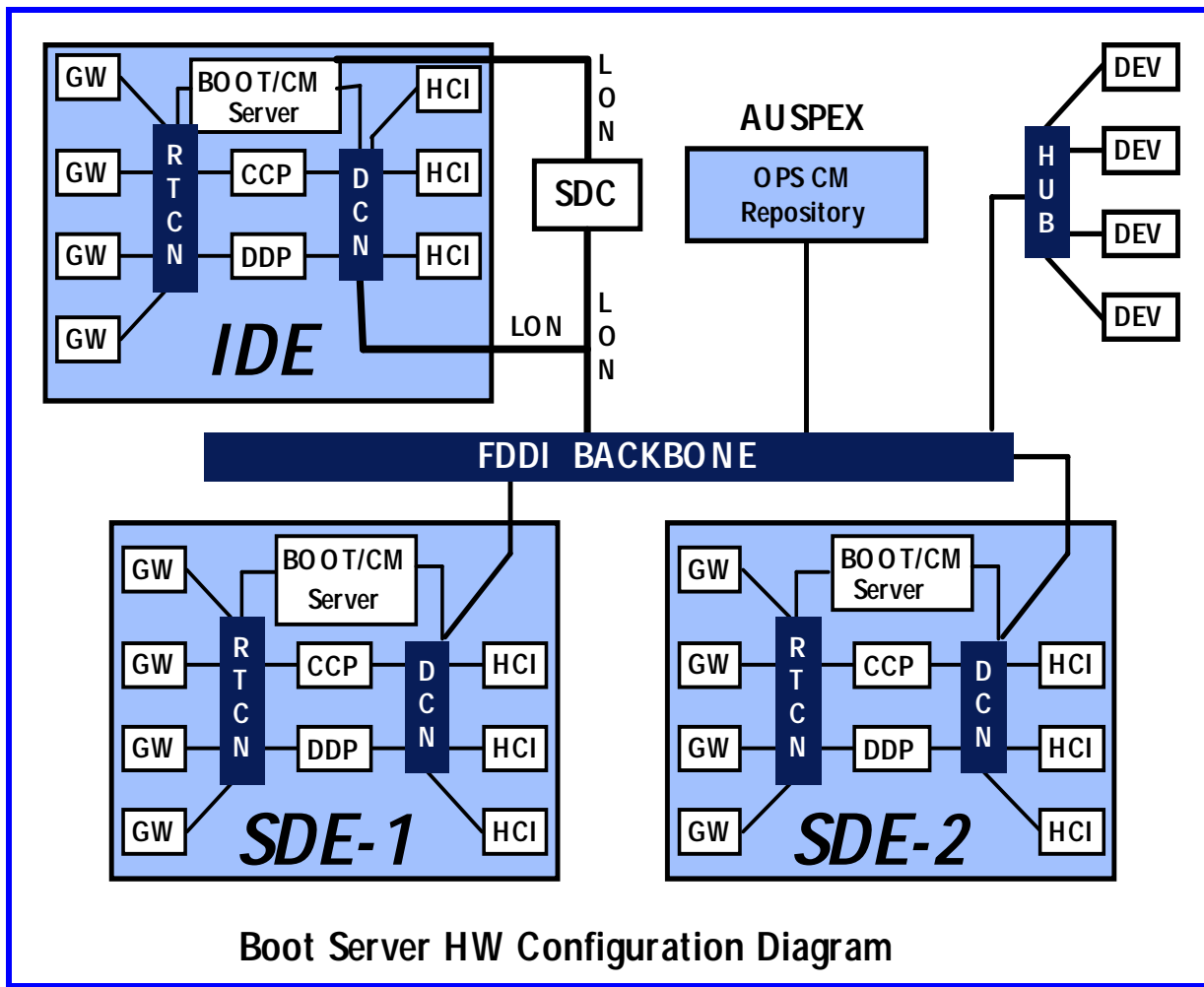
## 1.2.3 Operating System Performance Requirements

### 1. Server Configuration

- 1.1 Operating Systems Boot Server Configuration will facilitate the down load of any single client machine in a time period not to exceed 30 minutes.

Note: Initialization time ends when the Boot Server process initiates the client workstation start-up login screen for the HCI, CCP, and DDP.

### 1.2.4 Operating System Configuration



## 1.3 Operating System Design Specification

The Operating System design concept is generic to vendor platforms. The mechanisms and procedures to accomplish the load is specific to a vendor.

The first step of creating the Operating System is to create a Generic Build Image. This image contains all the products, configuration and tunable parameters that are generic to all loads. Then the Generic Image is customized for a particular platform such as graphic and network drivers.

The second step is to build a Boot server that will distribute all the images to the client machines. Once the Boot server is constructed, the generic image of a platform is placed on the server for distribution. For Thor, a 4MM DAT tape or disk drive will be the distribution media for the Boot server images. The distribution of client images is accomplished via a diskless fashion and then rebooted as a standalone system.

## Software Requirements and Design Specification Template

The final step is for the client workstation to boot from the Boot server. During this diskless booting, the client will identify and reconfigure itself as a standalone workstation. At this point the client workstation is ready to download custom code.



## 1.3.2 Operating System CSC External Interfaces

### 1.3.2.1 Operating System Message Formats

NA

### 1.3.2.2 Operating System Name Display Formats

NA

### 1.3.2.3 Operating System Input Formats

NA

### 1.3.2.4 Operating System Printer Formats

NA

## 1.3.3 Operating System Test Plan

OS Validation and Platform Operations testing will consist of procedures and utilities to test key components and aspects of each platform and function. The key areas include but are not limited to:

1. OS Baseline Software
2. Network Configuration and Connectivity
3. Platform Specific Functionality
4. Software Patch Levels
5. Control File Validation.

### 1.3.3.1 Test Environment

OS Validation and Platform testing will consist of processes common to all environments and localized tests which are unique to each specific environment. Common tests will be ones such as Software Comparator Tests to address areas which are intended to be common throughout the environment (SDE1, SDE2, CLCS, etc). Localized tests will address those areas which are necessarily unique (network addressing, server connectivity, etc).

### 1.3.3.2 Test Methods

Testing methods will be constructed to be performed in a non-intrusive manner. They will use utilities and commands readily available on each platform and be designed to minimize customization. Unix command-line testing will include standard Unix commands such as **diff**, **ping**, **ifconfig**, **grep**, and others.

### 1.3.3.3 Test Cases

#### 1.3.3.3.1 OS Baseline Software

Verification that the Operating System includes all products originally loaded will be performed by running the IRIX command “versions” with the output re-directed to a collector file. This file is then compared to a Baseline Master Software File located in the **/var/adm/docs** subdirectory. The Baseline Master Software

# Software Requirements and Design Specification Template

File is generated once the baseline is approved and is copied into the generic image prior to uploading onto the BootServer so that it will be included with each platform download process.

## 1.3.3.3.2 Network Connectivity

Testing network connectivity and operation is an integral component of the OS Test Suite that assures proper interaction of BootServer, Netserver, Client workstations. The tests utilize the ping, netstat, and ifconfig commands to verify correct routing and connectivity. While there is currently no performance criteria, if it becomes apparent there are performance constraints, those tests would be included in this Test Suite.

## 1.3.3.3.3 Platform Specific Functionality

Testing of Platform Functionality is currently being developed to include a database that allows reporting and verification of functional parameters associated with each workstation function. This Test Suite is primarily designed to be used at the BootServer to poll a workstation in question to verify proper setup of functions on that workstation. Future Platform Functionality tests to be developed will be aimed at methods of testing specific functions on the client.

## 1.3.3.3.4 Software Patch Levels

Verification of Software Patch Levels is performed by reporting installed patches on the workstation using the IRIX command “**versions | grep patch**”. It is a routine test, frequently used when researching software configuration issues.

## 1.3.3.3.5 Control File Validation

Control File Validation is primarily used when installing new clients and confirming setup of the diskless client and share trees on the BootServer. It is currently performed by issuing the command “**chkclient clcs55**” where **clcs55** is the name of the workstation in question. It is executed on the BootServer from the **/wsoscm2/DL** subdirectory and reports all the information specific to that client from the control files.



## Appendix A

---